CUSTIS®

Ответственность за качество в разных ИТ-проектах: в чем она и как ее разделять

Максим Цепков
Главный архитектор дирекции развития решений

Software quality assurance days (SQA days)
Минск, 25 ноября 2016

CUSTIS®

Тестировщик и качество: знаем ли мы смысл понятий?

- Многие полагают, что понятия однозначны и фразу «Проект надо сделать качественно, и тестировщик отвечает за это» все **должны** понимать одинаково
- На самом деле, мир меняется, а **смысл** понятий **меняется** вместе с ним
- В головах многих смысл остается таким, каким был в момент узнавания
- Мы поговорим о том, **какое** бывает качество проектов и **за что** отвечают тестировщики

Чтобы вы понимали весь спектр и это не стало неожиданностью в новом проекте или компании

CUSTIS®

План доклада

- Какое качество нужно разным проектам?
  - Как менялись представления о «качественном проекте»
  - Какое качество нужно для разной ИТ-разработки
- Качество: кто и за что отвечает?
  - Между кем разделяется ответственность
  - Как работать с границами ответственности и какова ответственность тестировщика
- А как все-таки меняются смыслы?
  - Team: как нам понимать друг друга и эффективно сотрудничать

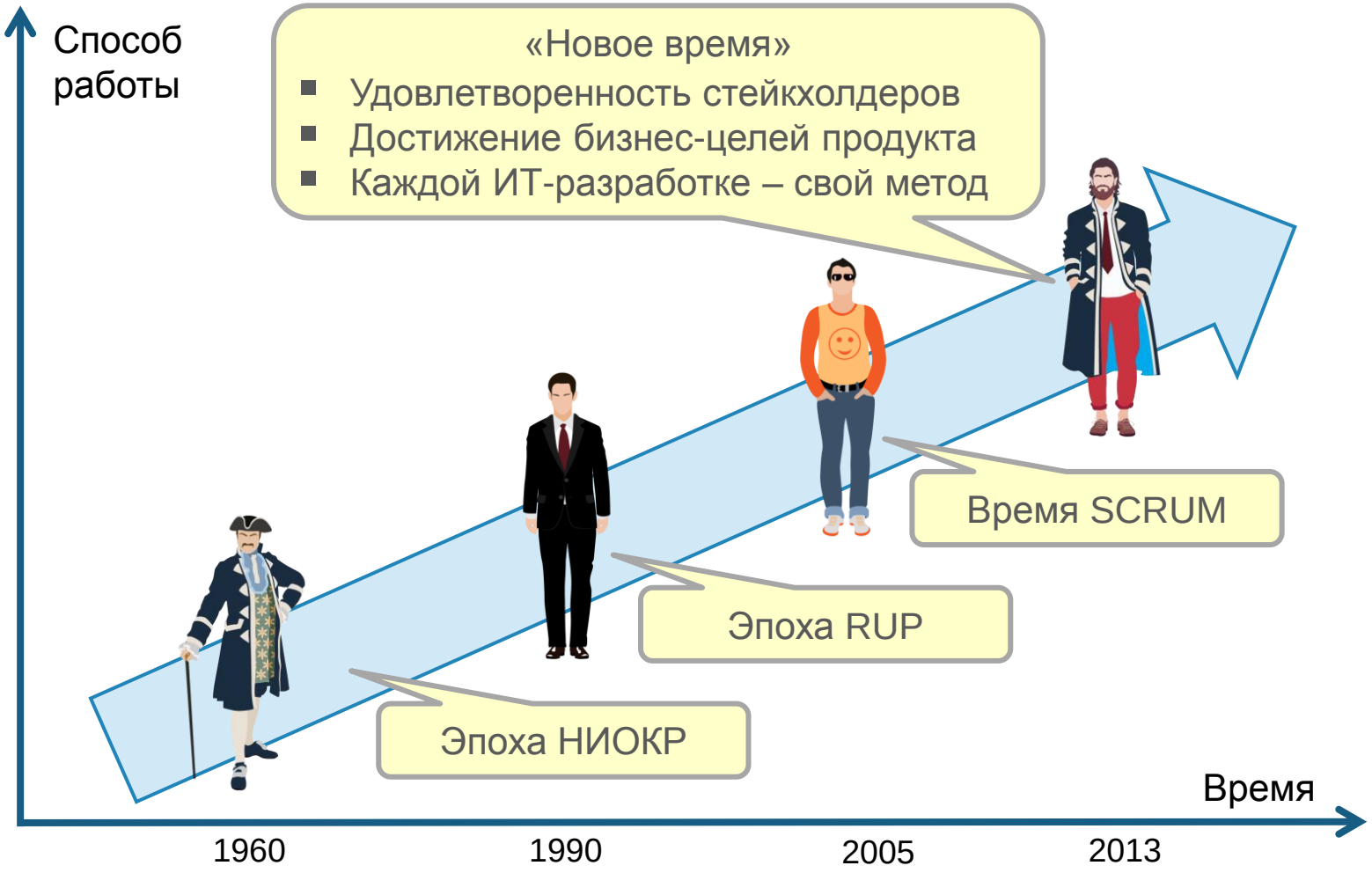
CUSTIS®

Как менялись представления о «качественном проекте»

По мотивам доклада «Развитие критериев качества и управления проектами» на AgileDays – 2015

CUSTIS®

Big Picture истории развития



CUSTIS®

Эпоха НИОКР: когда компьютеры были большими

- Создавались большие и сложные проекты
- Требования к проектам редко менялись
- Проекты делал квалифицированный персонал
- Упор был на **качество ИТ-системы**

Ф. Брукс
«Мифический человеко-месяц»

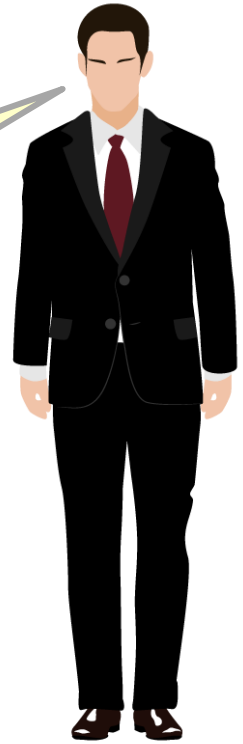
Цель проекта – создать **совершенную ИТ-систему** в **одном** экземпляре

CUSTIS®

Эпоха RUP: персоналки потребовали много разработчиков

- Применим к ИТ-разработке принципы промышленного производства
- Разделим задачу на этапы: проектирование, разработка, внедрение
- Наладим процессы и разделим зоны ответственности

PMBOK-3 (2004)
RUP (2003)



Оценка качества: удалось ли выполнить проект в срок, уложиться в бюджет и достичь ожидаемых результатов

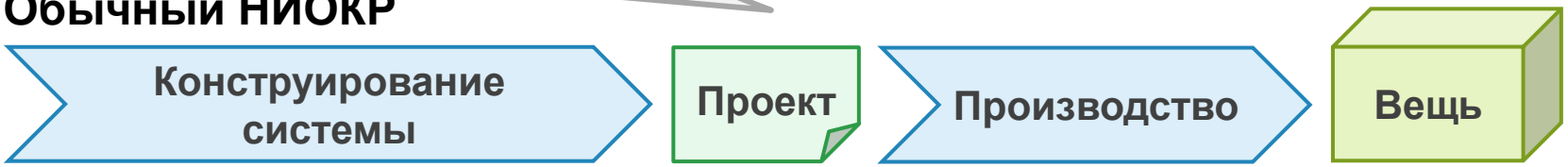
Получалось не очень

CUSTIS®

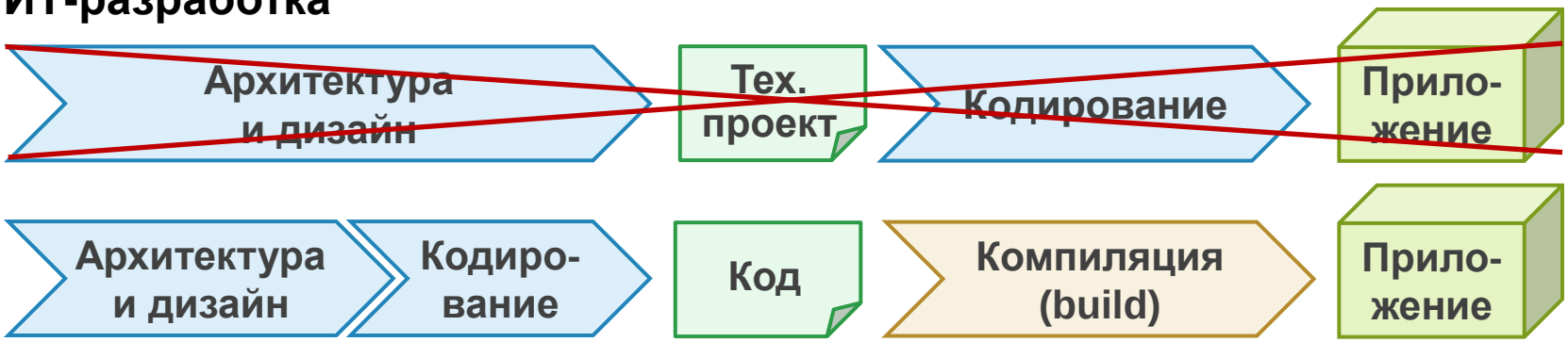
Природа ИТ мешает процедурам

Jack W. Reeves «What is software design» (1992; перевод)

Обычный НИОКР



ИТ-разработка

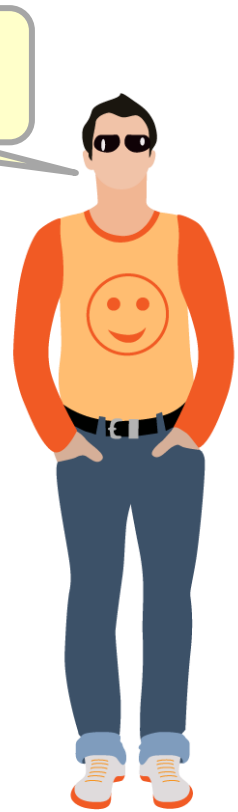


Несмотря на тяжелые и дорогие процедуры, успешность проектов была низкой. Не получалось обеспечить гибкость, а **успех определялся людьми**, которых не хватало

CUSTIS®

Agile и SCRUM: ответ на вызовы

- Вместо тщательного планирования – наблюдение за траекторией движения проекта и приближением к цели
- Концепция SMART-целей, измеримость достижения
- Итеративное движение с корректировкой положения цели (требования к системе)



Качественный проект – это частые инкрементальные поставки нужного софта

А что сейчас: векторы развития

- От проектной деятельности – к непрерывному развитию продукта
- От качества ИТ-системы – к удовлетворенности стейкхолдеров
- От создания системы – к достижению возможностей для бизнеса и пользователя
- Каждой ИТ-разработке – свой метод

Канбан в ИТ (2010)
DevOps (2012)

PMBOK 5 (2013)
частично

Качественная ИТ-разработка удовлетворяет стейкхолдеров и обеспечивает возможности для бизнеса

CUSTIS®

Энтони Лаудер «Культуры программных проектов»

- Моя схема сильно похожа на схему четырех культур Энтони Лаудера
  - Научная
  - Заводская
  - Дизайнерская
  - Сервисная
- Содержание уровней отличается

Оригинал, перевод (pdf), рецензия Стаса Фомина

Если схема Лаудера вам подойдет больше, используйте ее. Нет смысла выяснять, какая правильнее

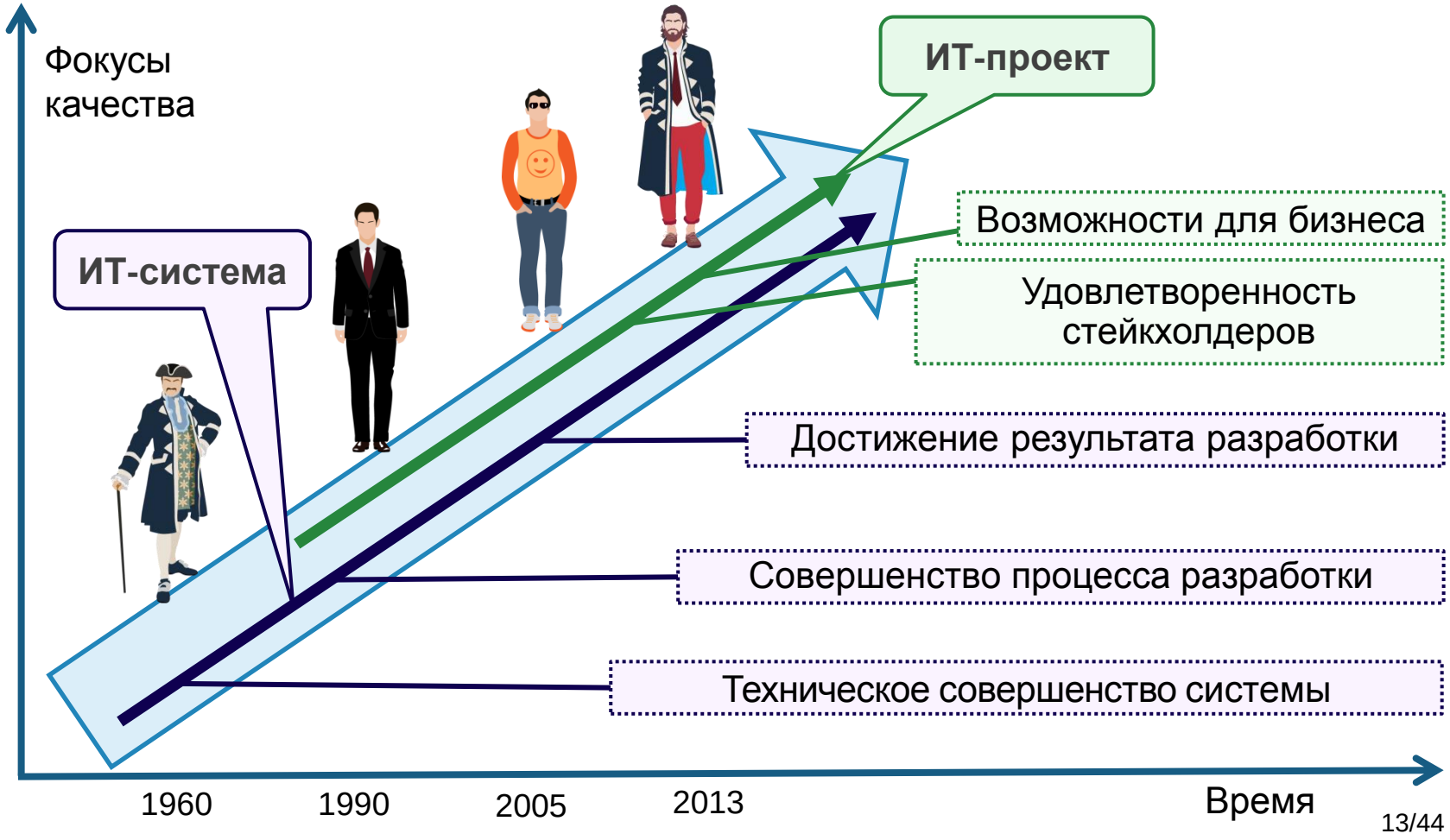
CUSTIS®

Какое качество нужно для разной **ИТ-разработки?**

Слово **«проект»** исчезло не случайно.
Меняем не только смысл, но и понятие!
Вместо «проекта» – **предприНятие, endeavor**

CUSTIS®

Каждый этап добавлял новые фокусы качества, а не отменял старые

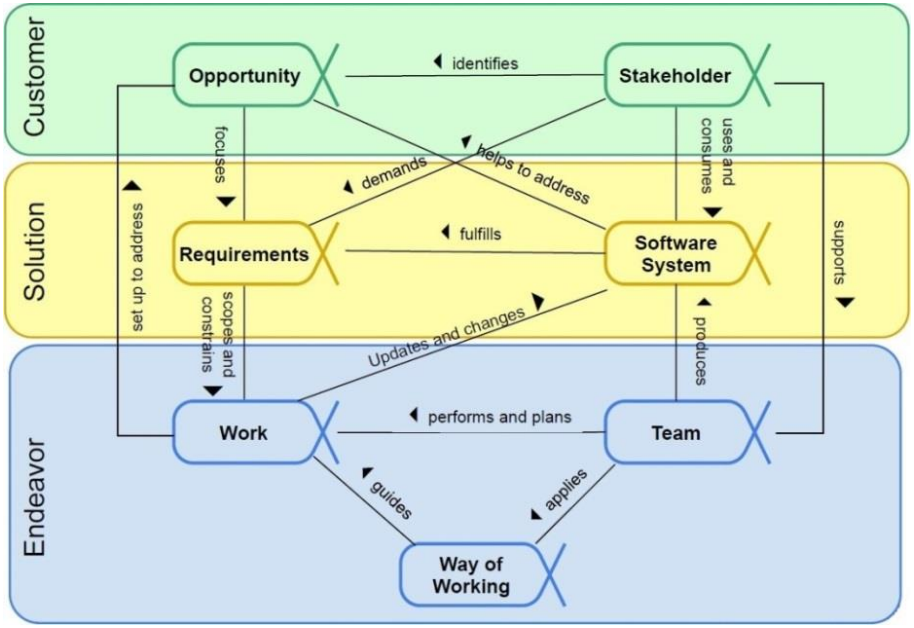


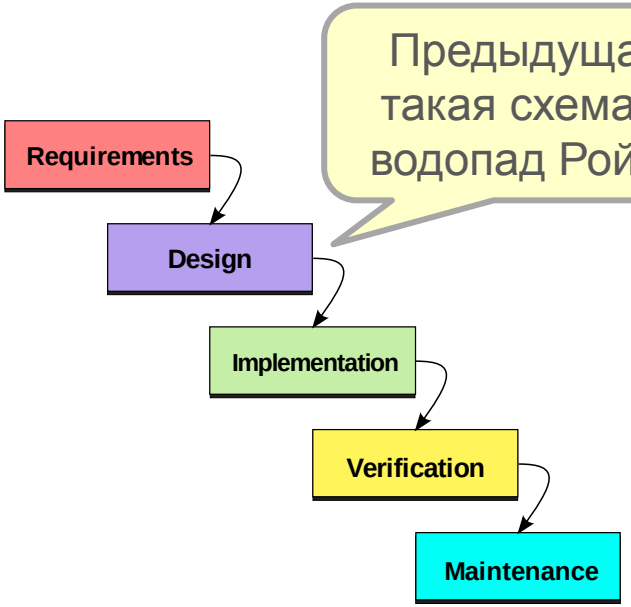
CUSTIS®

OMG Essence – пространство привязки фокусов качества

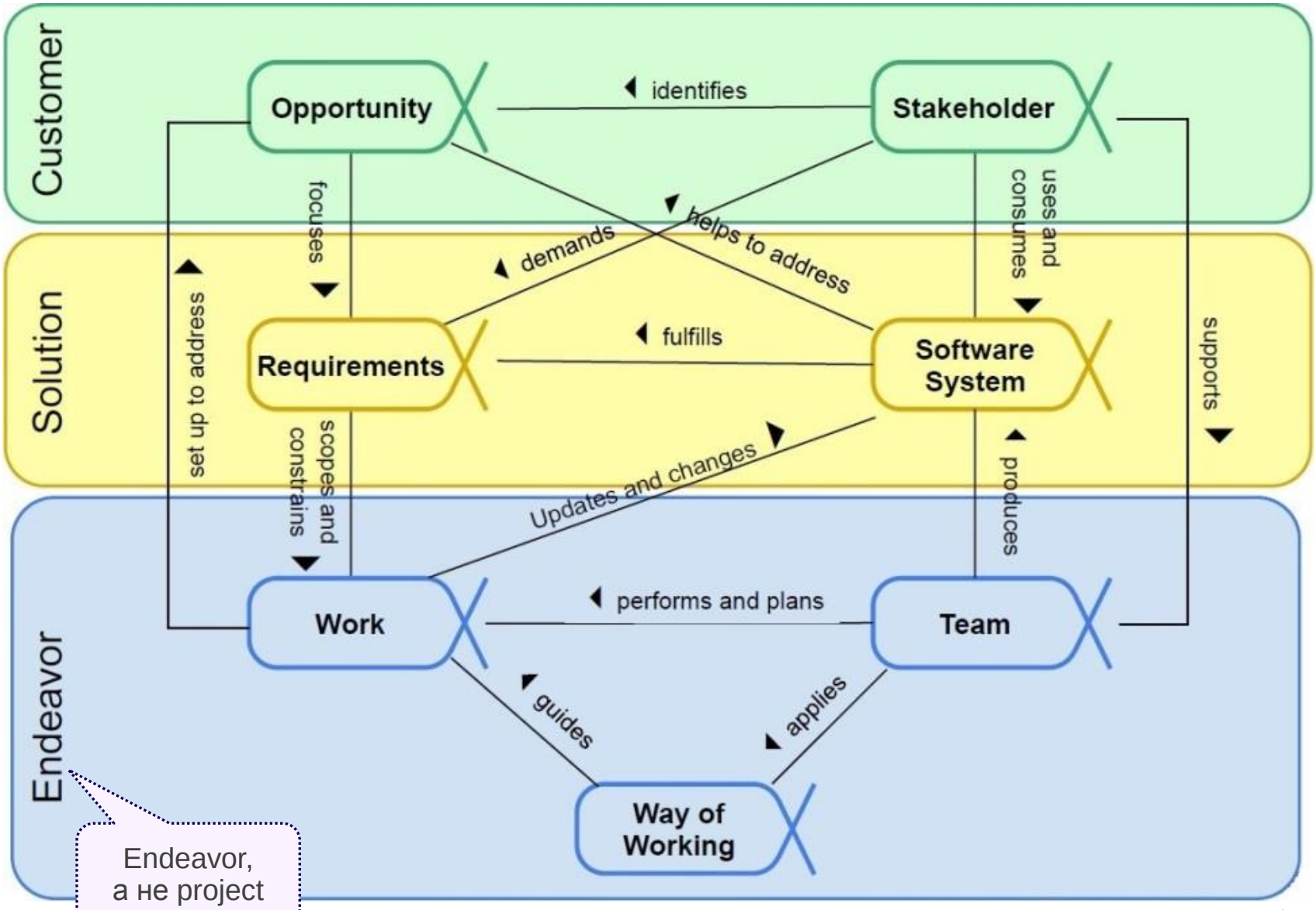
Создан группой SEMAT под руководством И. Якобсона

- Стандарт на сайте OMG
- Конкретные описания на сайте И. Якобсона
- Книга И. Якобсона The Essence of Software Engineering
- Курс системного мышления А. Левенчука (pdf)



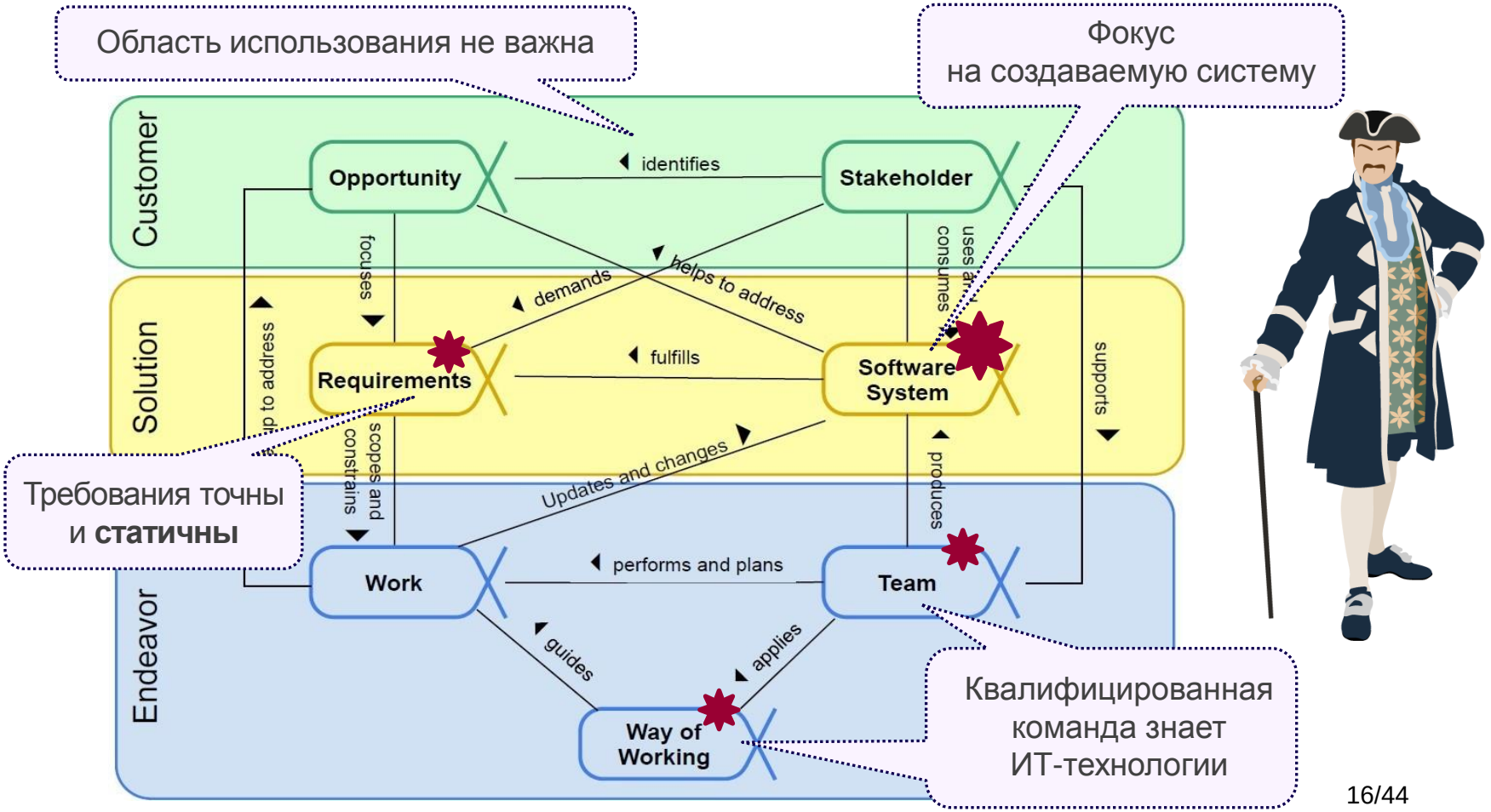


Предыдущая такая схема – водопад Ройса


Customer
Solution
Endeavor
Opportunity
identifies
Stakeholder
focuses
demands
helps to address
uses and consumes
Requirements
fulfills
Software System
set up to address
scopes and constrains
Updates and changes
produces
supports
Work
performs and plans
Team
guides
applies
Way of Working
Endeavor, а не project

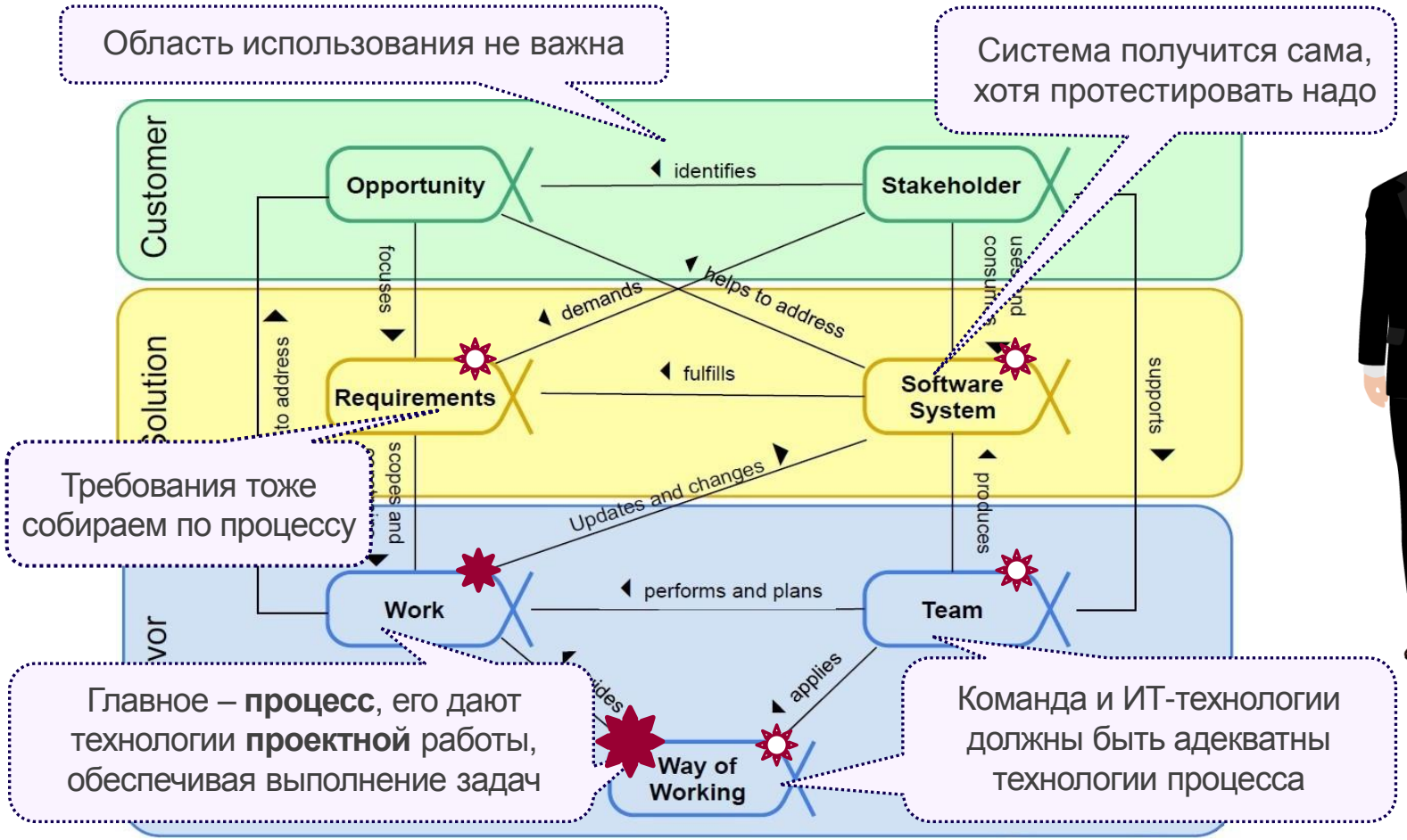
CUSTIS®

Система понятна как **черный** ящик, но **сложно** устроена – НИОКР



CUSTIS®

Система понятна как **белый** ящик – организуем **процесс** разработки

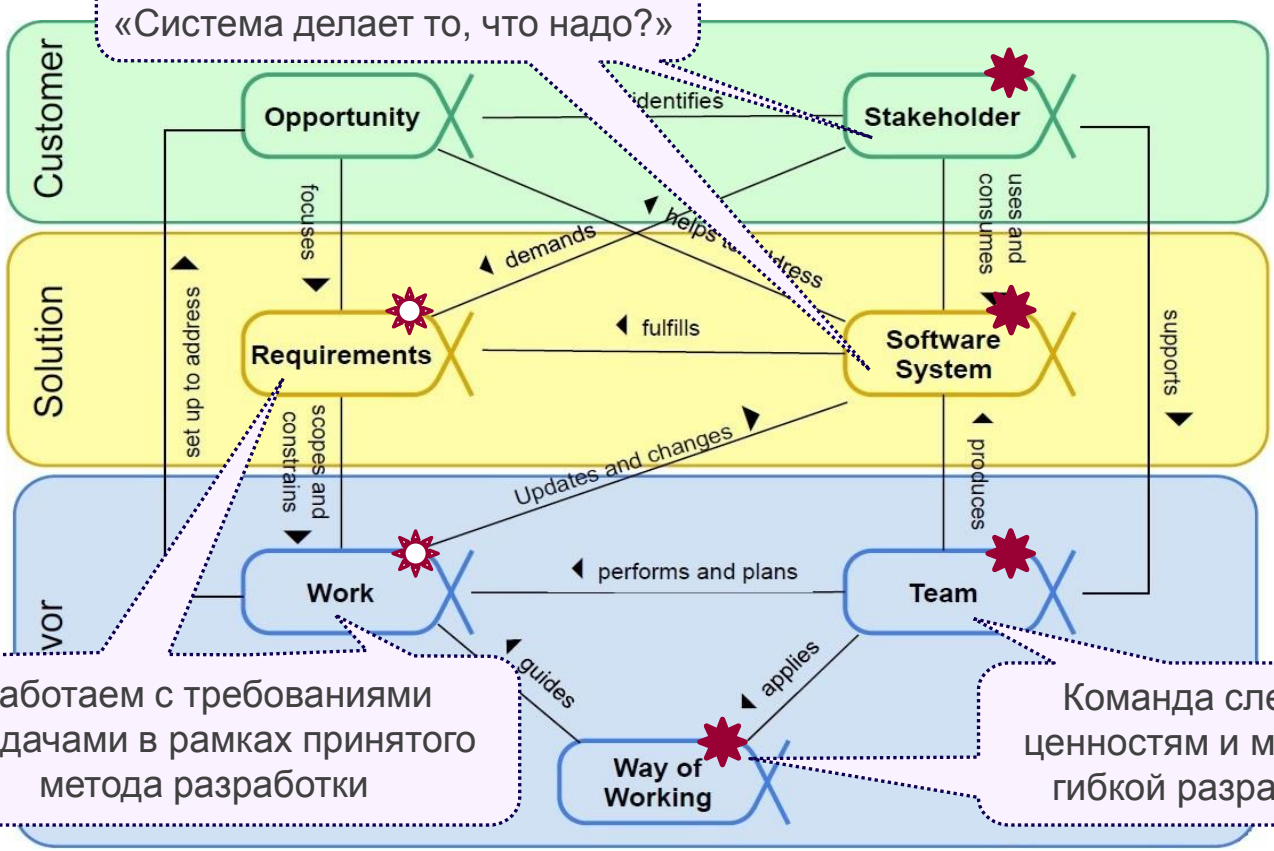


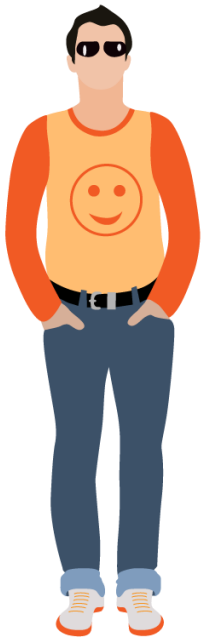
CUSTIS®

Образ системы неясен – приближаемся к нему итерациями

Объективно или из-за ограниченной квалификации команды

Постоянная проверка у стейкхолдеров: «Система делает то, что надо?»





Работаем с требованиями и задачами в рамках принятого метода разработки

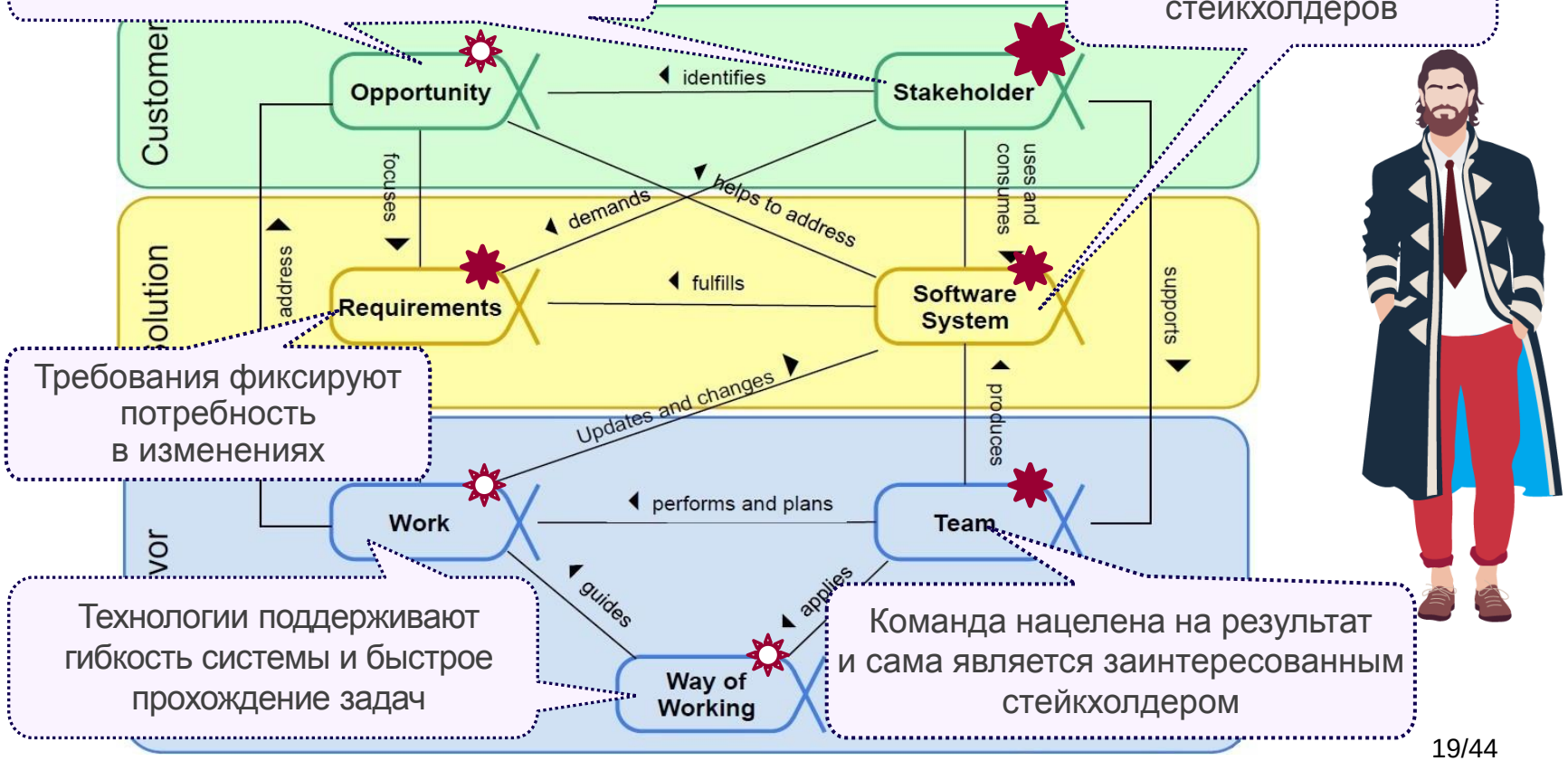
Команда следует ценностям и методам гибкой разработки

CUSTIS®

Непрерывная эволюция системы вслед за изменениями окружения

Стейкхолдеры требуют изменений системы, исходя из изменений в мире

Система быстро изменяется, удовлетворяя стейкхолдеров



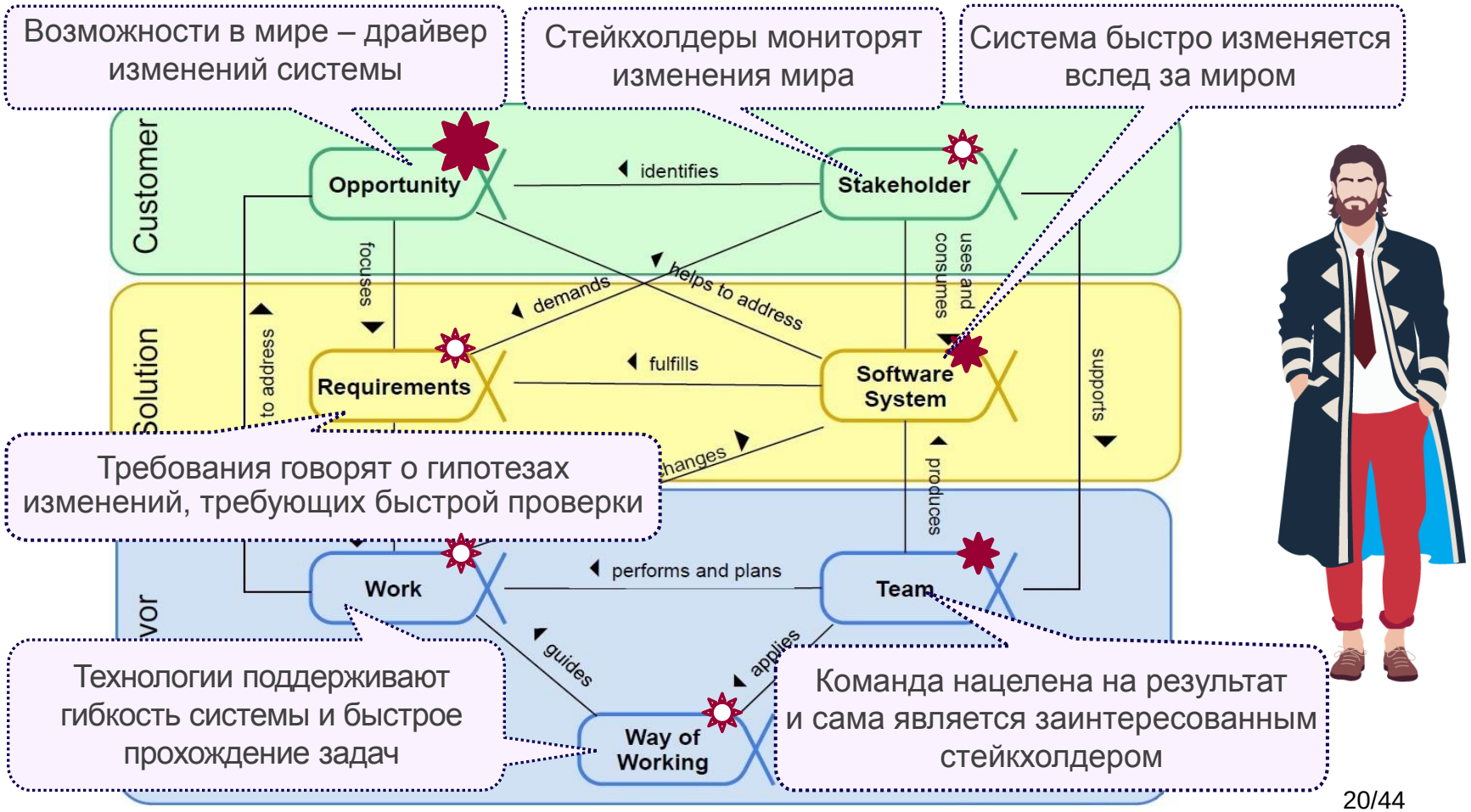
Требования фиксируют потребность в изменениях

Технологии поддерживают гибкость системы и быстрое прохождение задач

Команда нацелена на результат и сама является заинтересованным стейкхолдером

CUSTIS®

Система обеспечивает ожидаемые возможности развития бизнеса



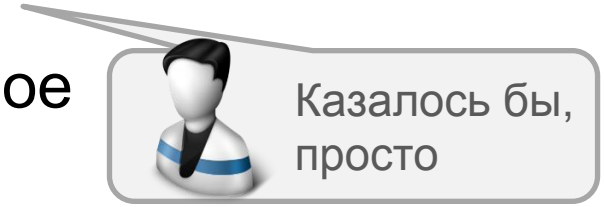
CUSTIS®

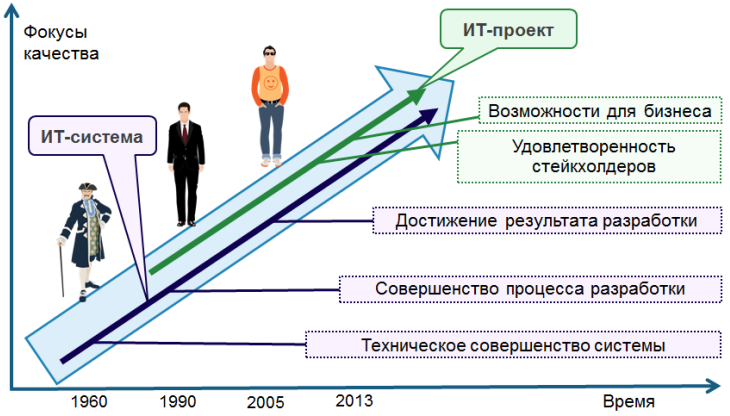
Между кем разделяется ответственность

Развитие доклада «Разделение ответственности в заказной разработке» на AnalystDays – 2015

QE и QA – послание от RUP

- Quality Engineer отвечает за **качество ИТ-системы** и проверяет ее тестированием
- Quality Assurance отвечает за **качество процесса**, которое в замысле должно привести к качеству системы
- Это лишь **два фокуса** ответственности из многих
- Остальные могут быть возложены на тех же людей или на отдельных лиц

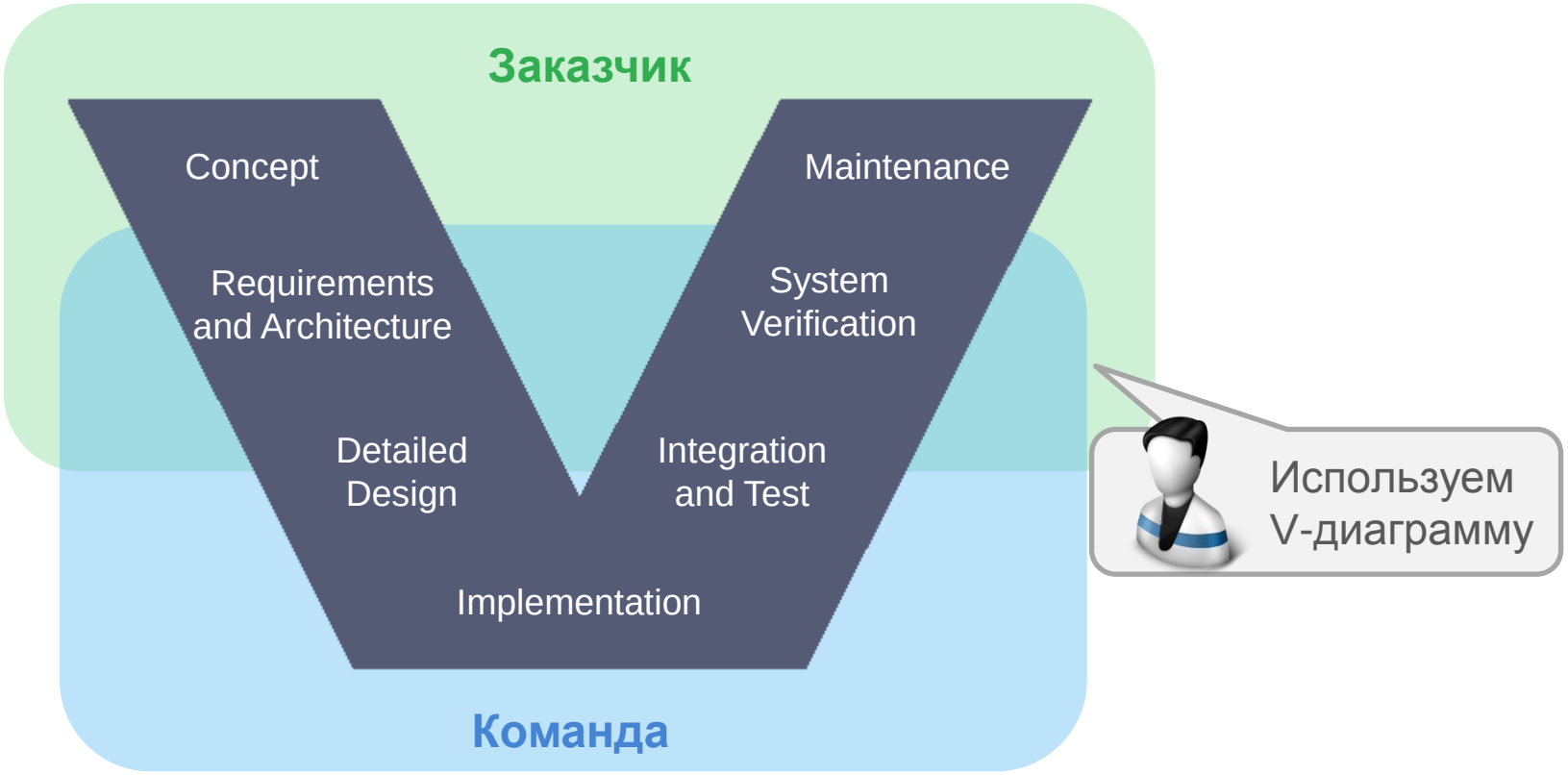




CUSTIS®

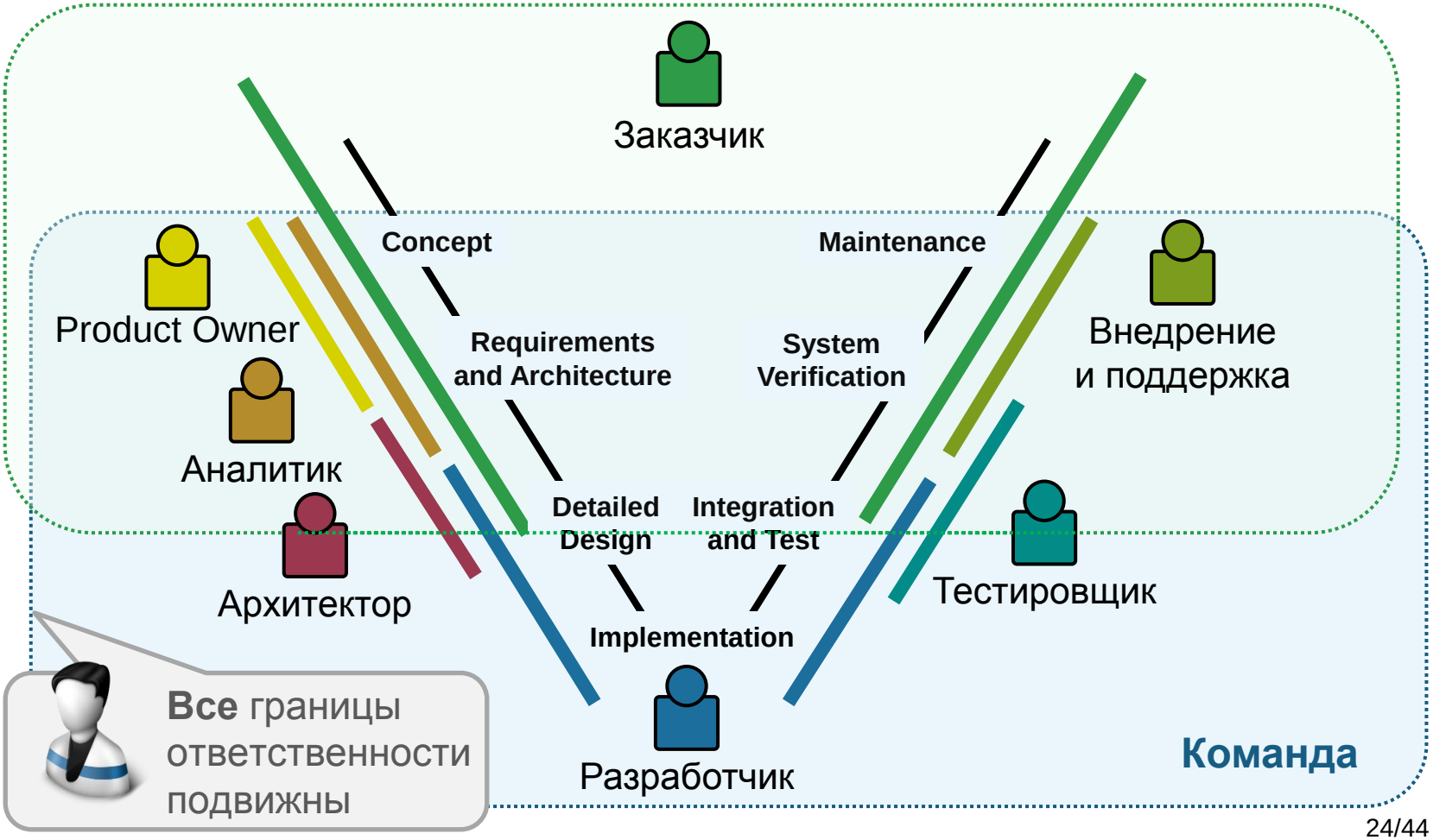
Кто действующие лица?

Прежде чем говорить об ответственности,
следует понять, **между кем** она распределяется



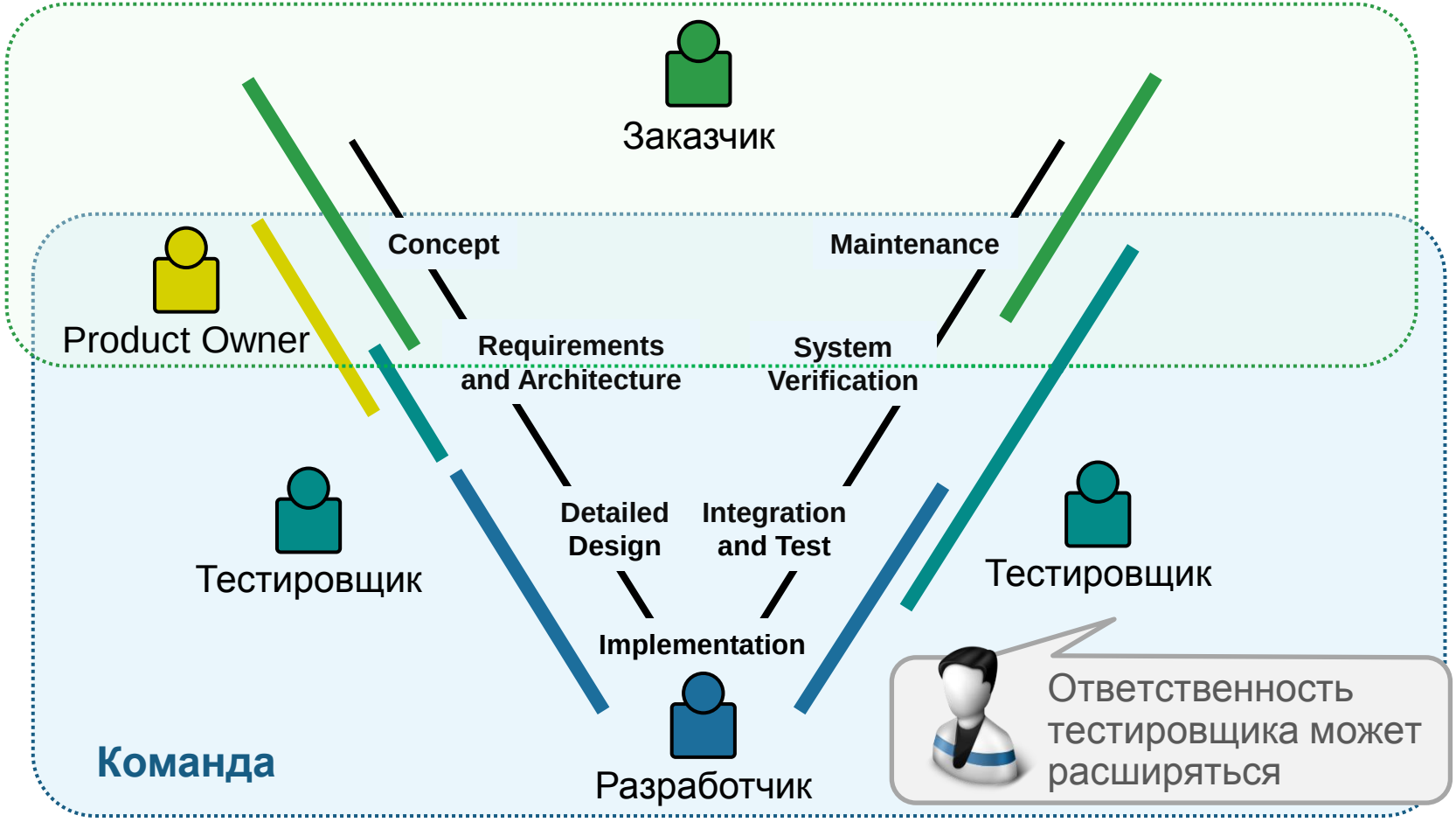
CUSTIS®

Команда включает много ролей



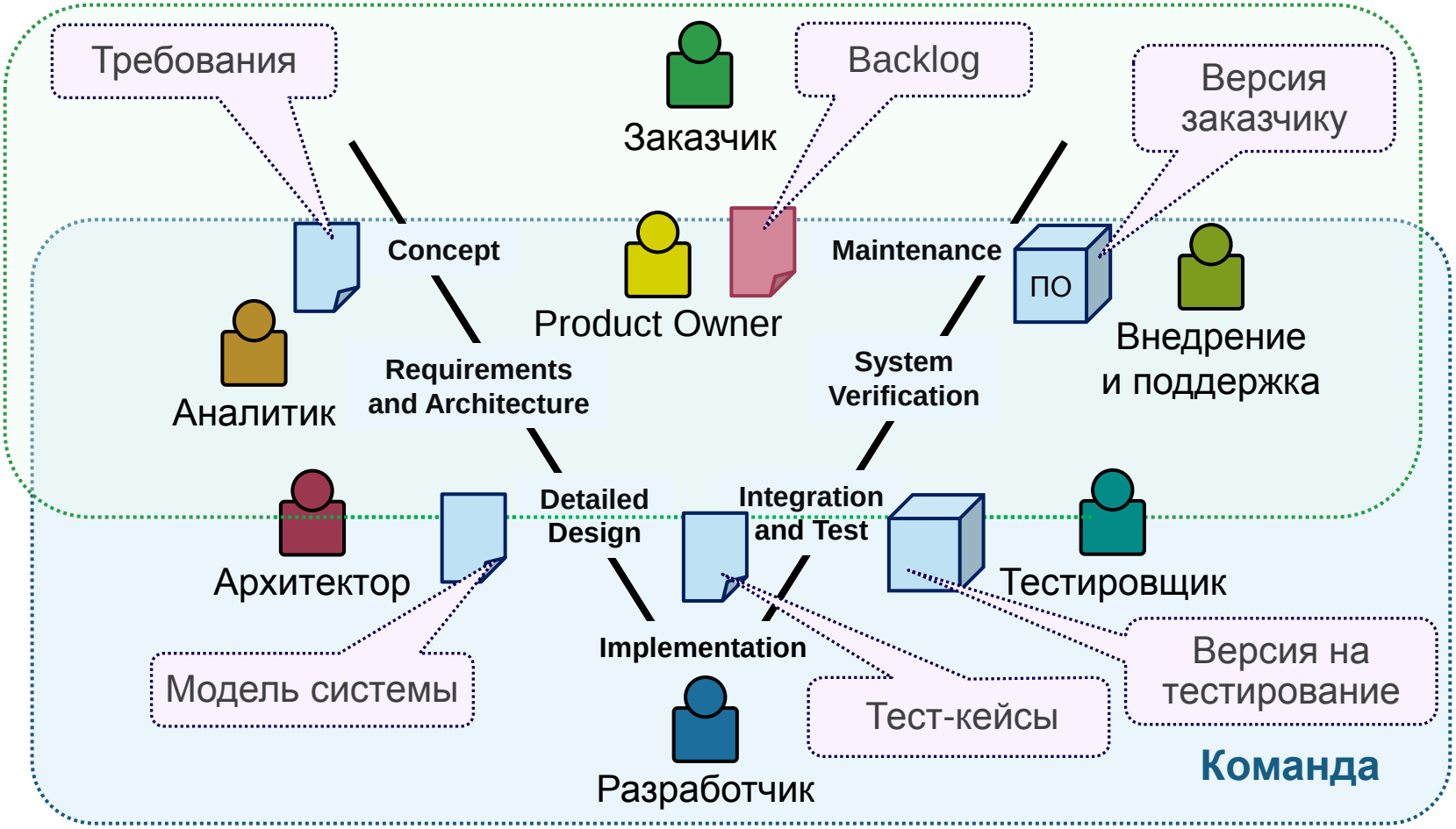
CUSTIS®

Не все роли могут присутствовать, а область заказчика может быть меньше



CUSTIS®

А еще нужно делить ответственность за артефакты



CUSTIS®

Как работать с границами ответственности и какова ответственность тестировщика

CUSTIS®

Используем Lifecycle OMG Essence

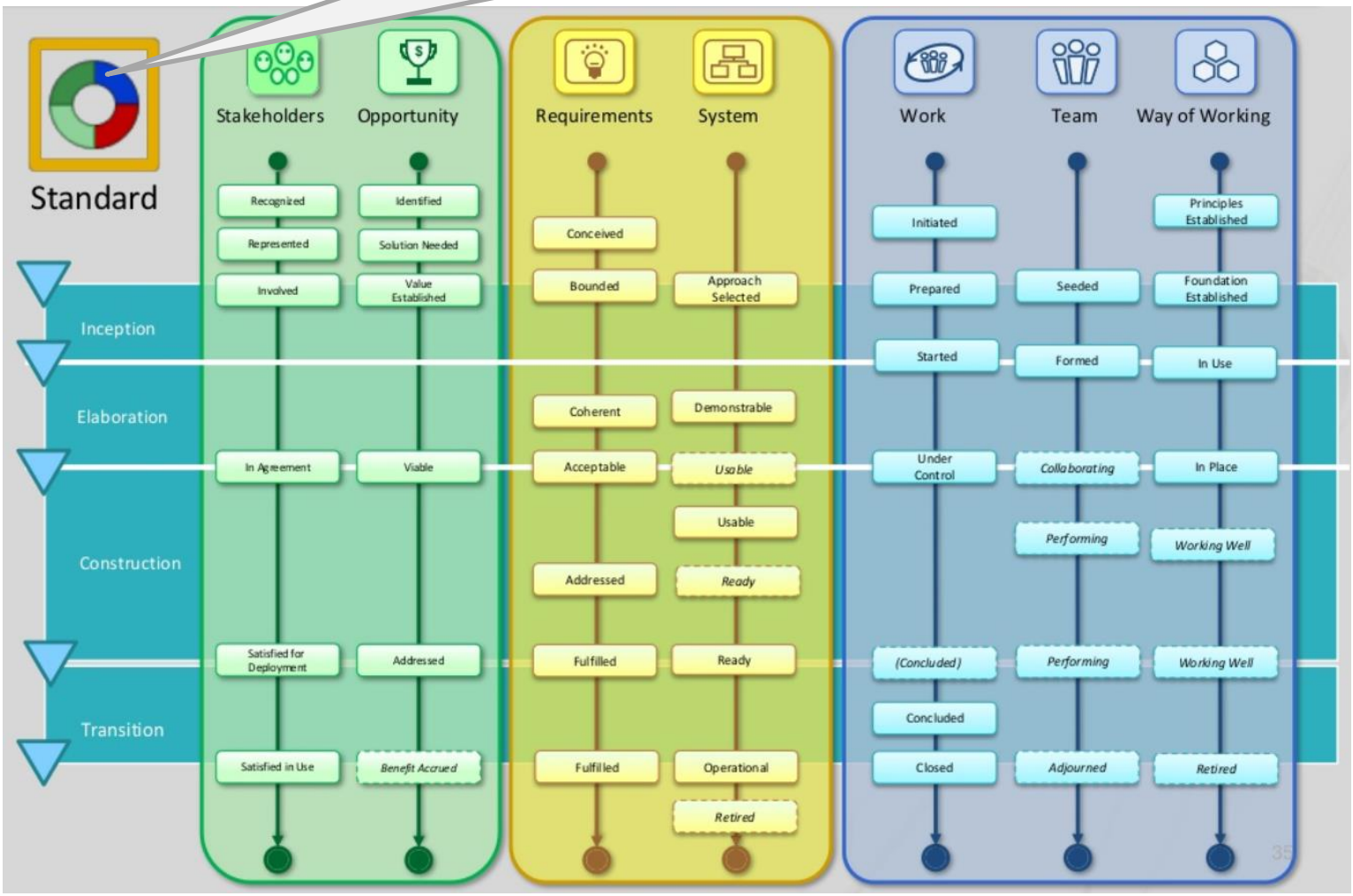
- Альфы задают объекты, состояния которых отражают движение проекта

Артефакт – частный вид объекта

- Ответственность делится по этим объектам
- Check lists состояний определяют, в чем ответственность
- Lifecycle-диаграмма иерархична: можно представить весь проект, его релиз, спринт или разработку фичи

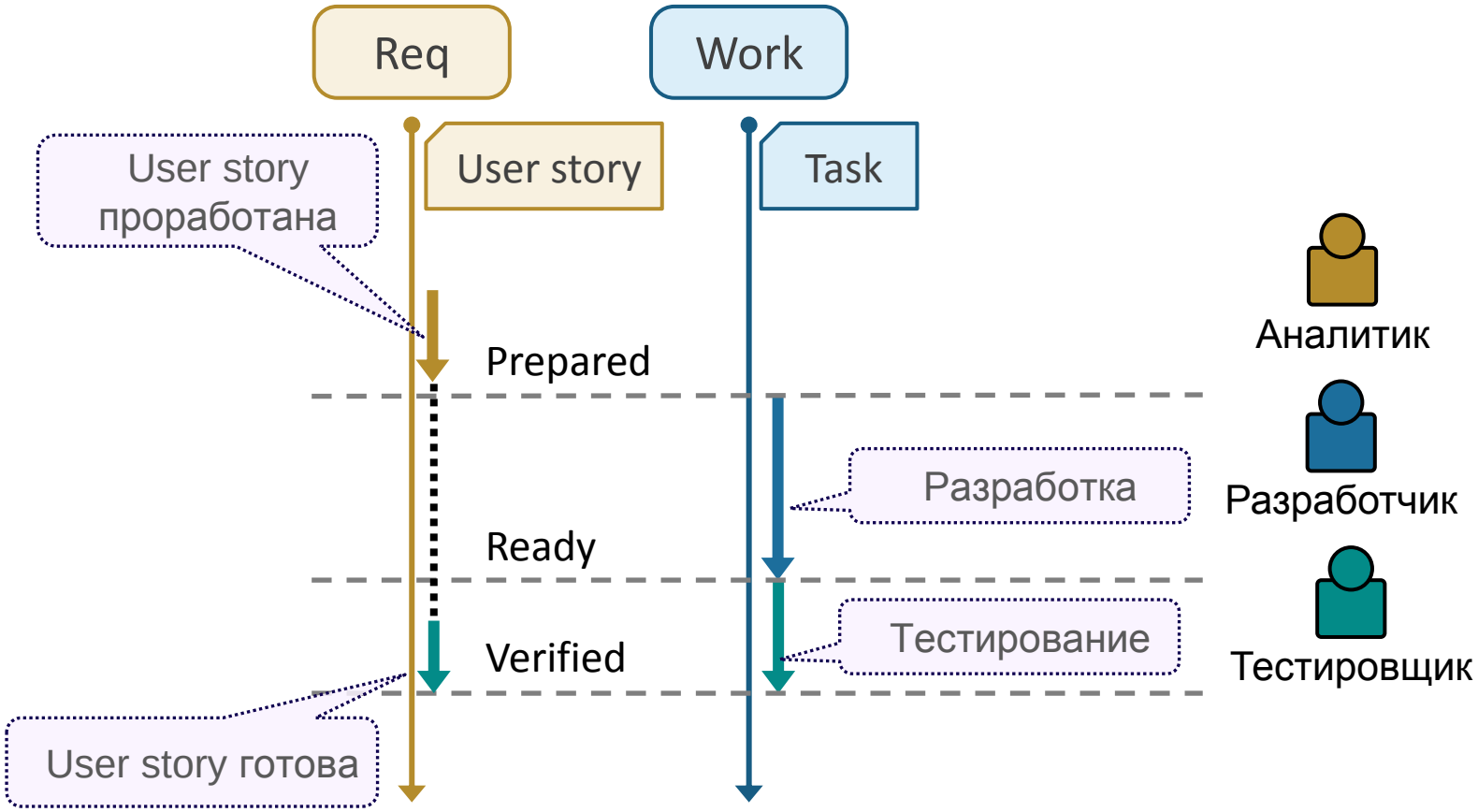
CUSTIS®

Пример: жизненный цикл «стандартного» проекта



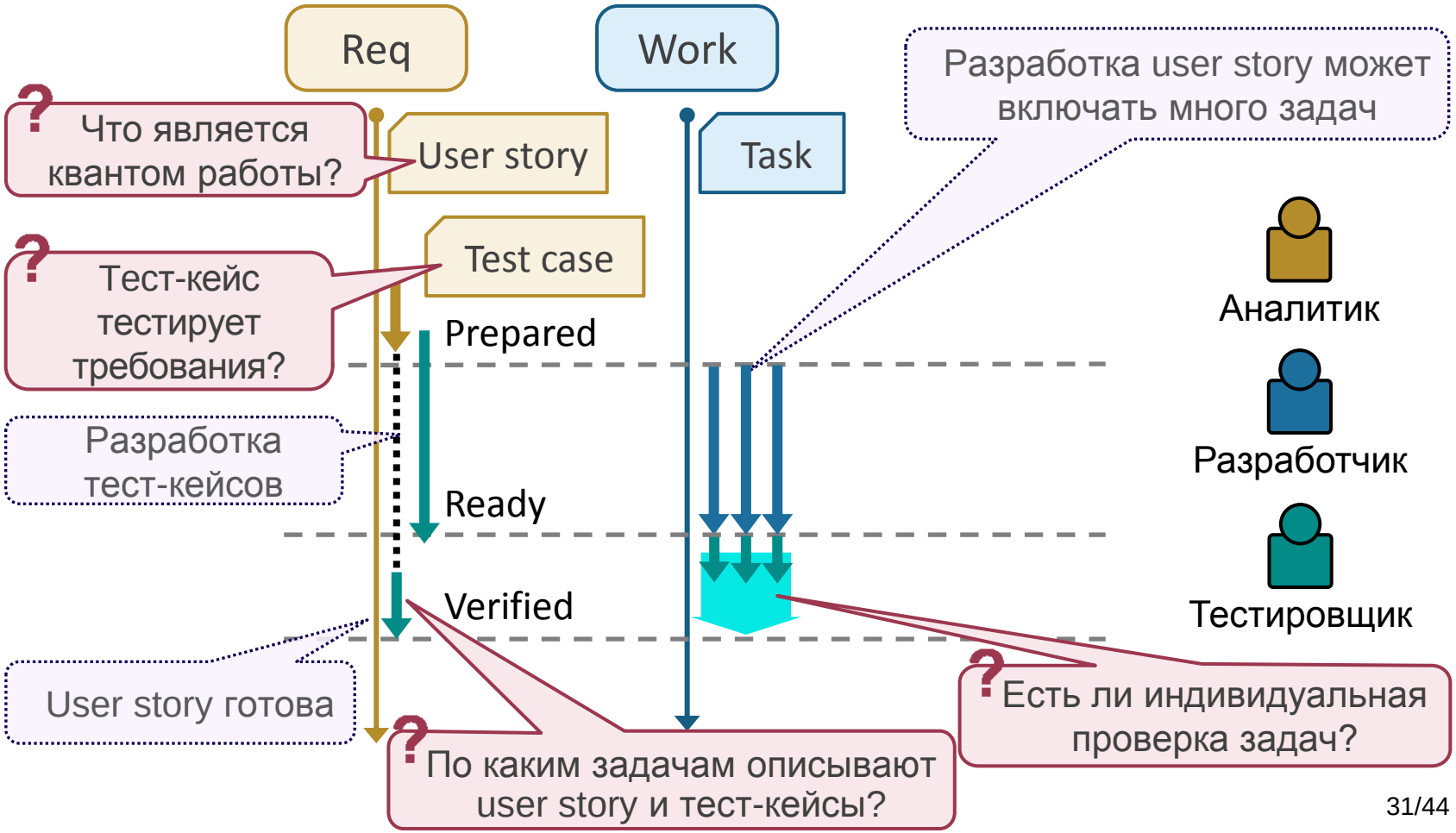
CUSTIS®

Простейший вариант – ответственность за проверку задачи



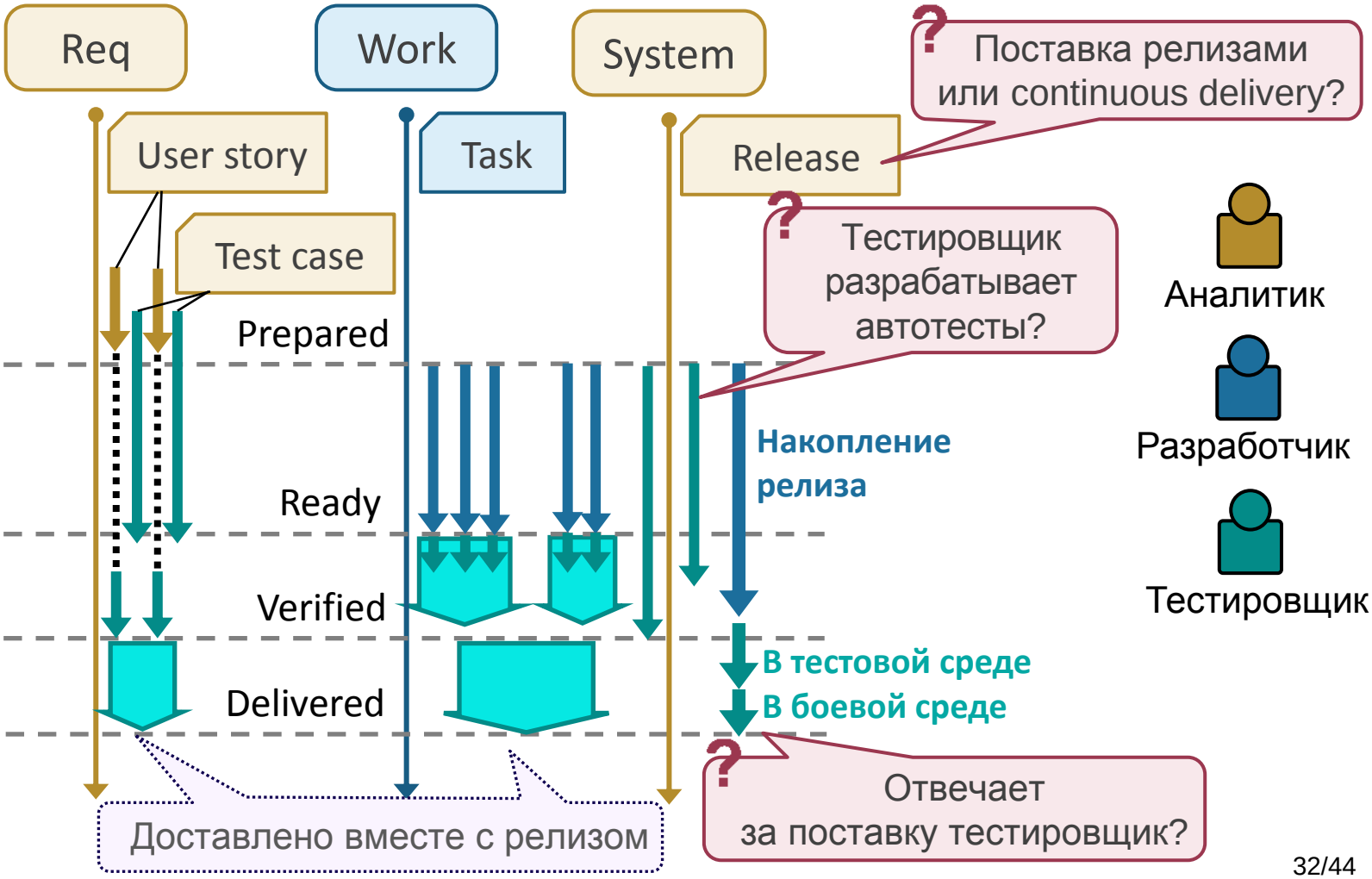
CUSTIS®

Ответственность за готовность user story



CUSTIS®

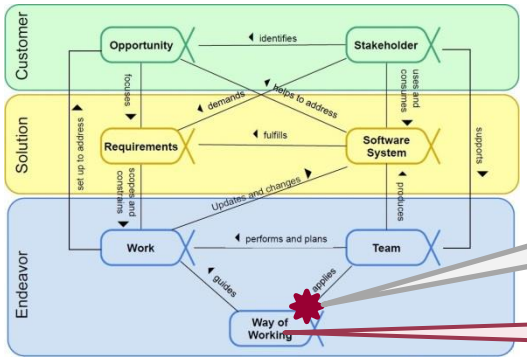
Ответственность за поставку системы



CUSTIS®

Что значит «может быть так»? Нужно ли иначе?

- Этот способ работы в вашей разработке сложился исторически или был спроектирован?
- Способ был адекватен особенностям разработки? Какие цели он обеспечивает?
- Способ продолжает оставаться адекватным сейчас или его пора заменять? Чем его заменять?



Это вопросы к альфе технологий, **Way Of Working**

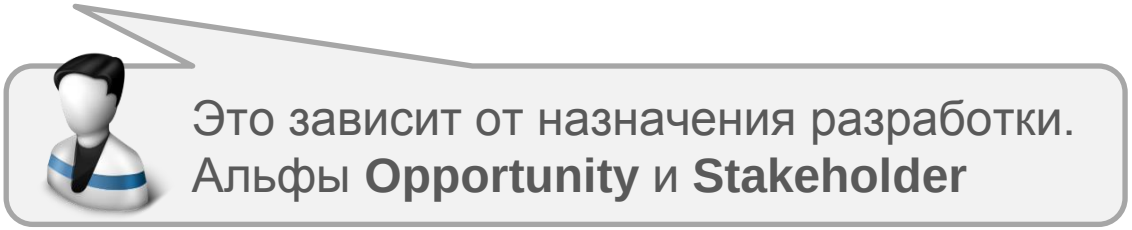
?

Как тестировщики участвуют в формировании технологий?

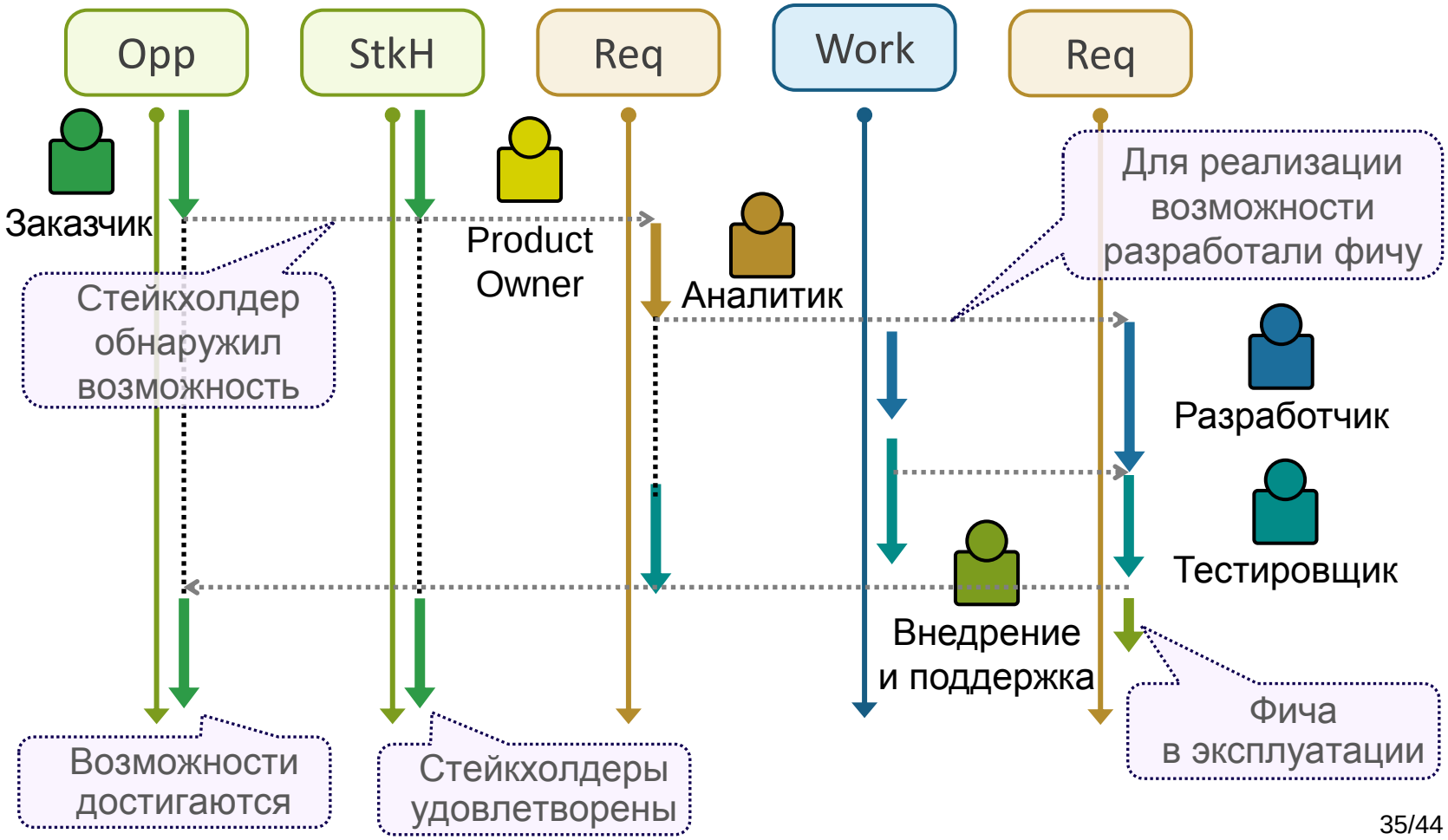
CUSTIS®

Как определять технологии?

- Нужна ли разработке continuous delivery или уместна поставка релизов?
- Нужны ли разработке автотесты и если нужны, то в каком объеме?

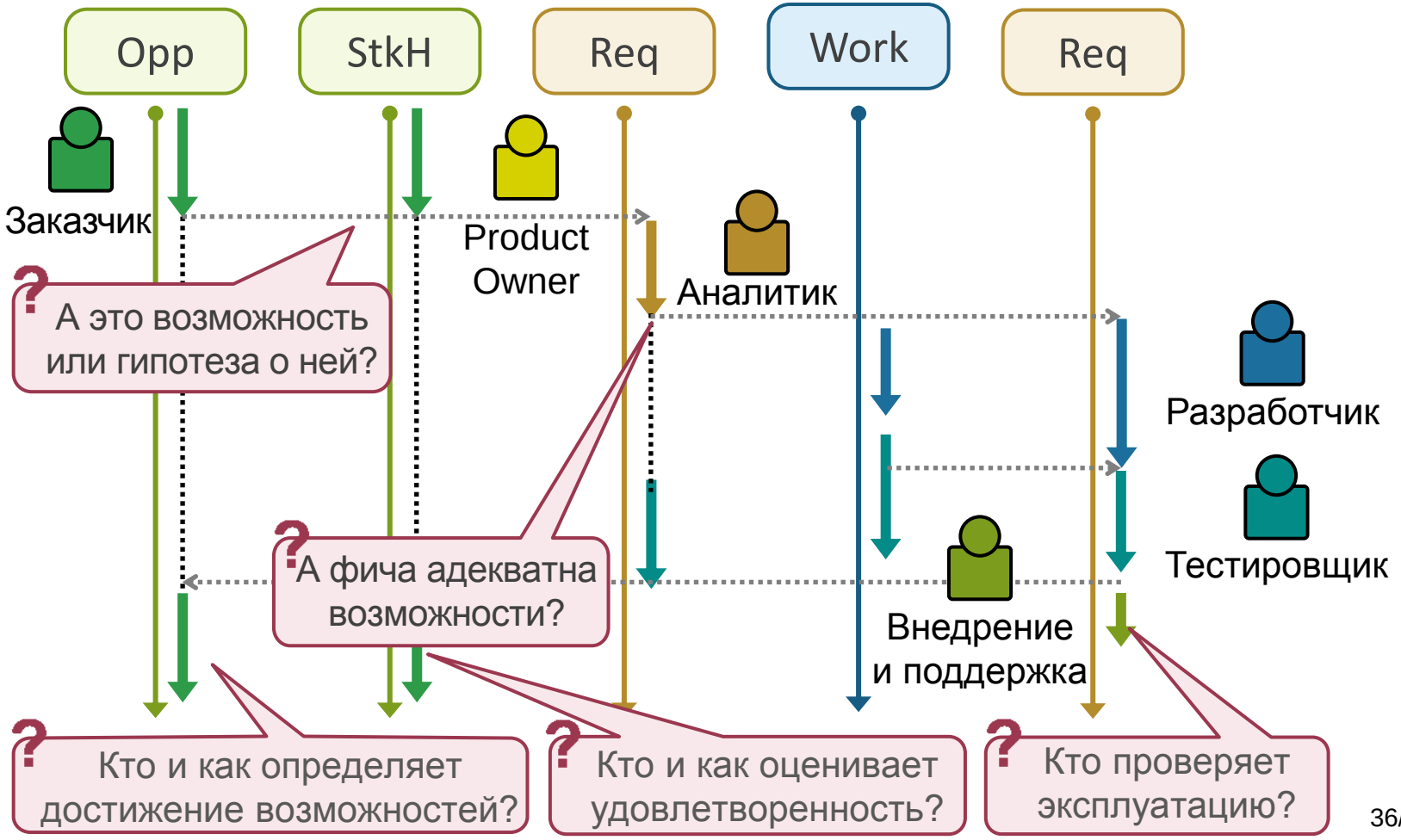


Удовлетворенность стейкхолдеров и обеспечение возможностей бизнеса



CUSTIS®

А кто и как проверяет цели и их достижение?



CUSTIS®

Ответственность за возможности

- Заказчик отвечает за опознание возможности или выдает гипотезы?
- Могут ли стейкхолдеры заказчика оценить проект фичи на соответствие возможности?
- Проявляют ли стейкхолдеры заказчика возможности для бизнеса в своих запросах?

CUSTIS®

Если возможности – лишь гипотезы или нет гарантии их достижения

- Необходимо определить ответственность и способ оценки гипотез
- Предпочтительно continuous delivery для быстрого цикла реализации
- Полезно A/B-тестирование
- Полезно применять тестирование гипотез до реализации или с помощью макетов

A/B-тестирование и проверка на макетах может входить в обязанности тестировщика, а может выполняться аналитиком или маркетологом

CUSTIS®

Как релиз приходит к пользователям?

- Можно ли автоматически проверить критичный функционал?
- Что тестируем автоматически?
- Отгружает релиз человек или автомат?
- Можно ли отменить отгрузку?

Continuous delivery

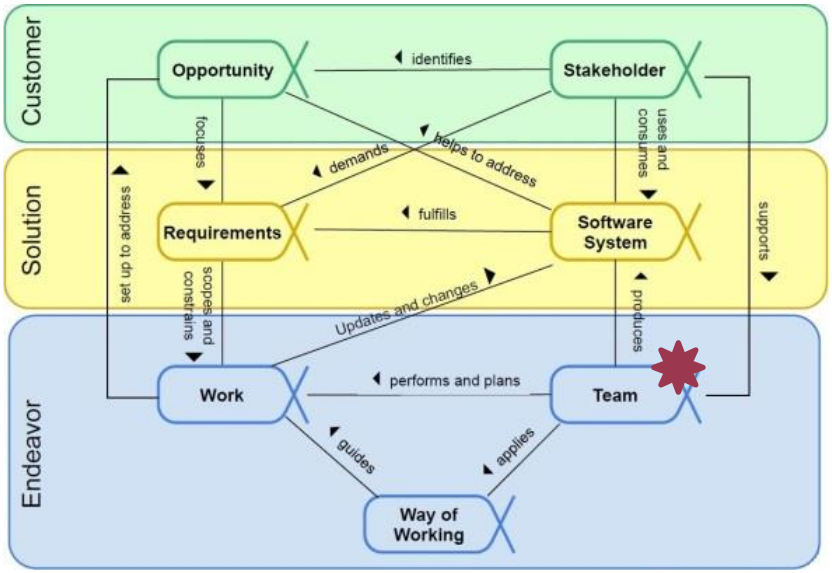
- Кто пишет новости версии?
- Эксплуатация – отдельная команда?
- Кто обучает пользователей?

DevOps

Область ответственности тестировщика и способ его работы сильно зависят от ответов на эти вопросы

CUSTIS®

Team: как нам понимать друг друга и эффективно сотрудничать



CUSTIS®

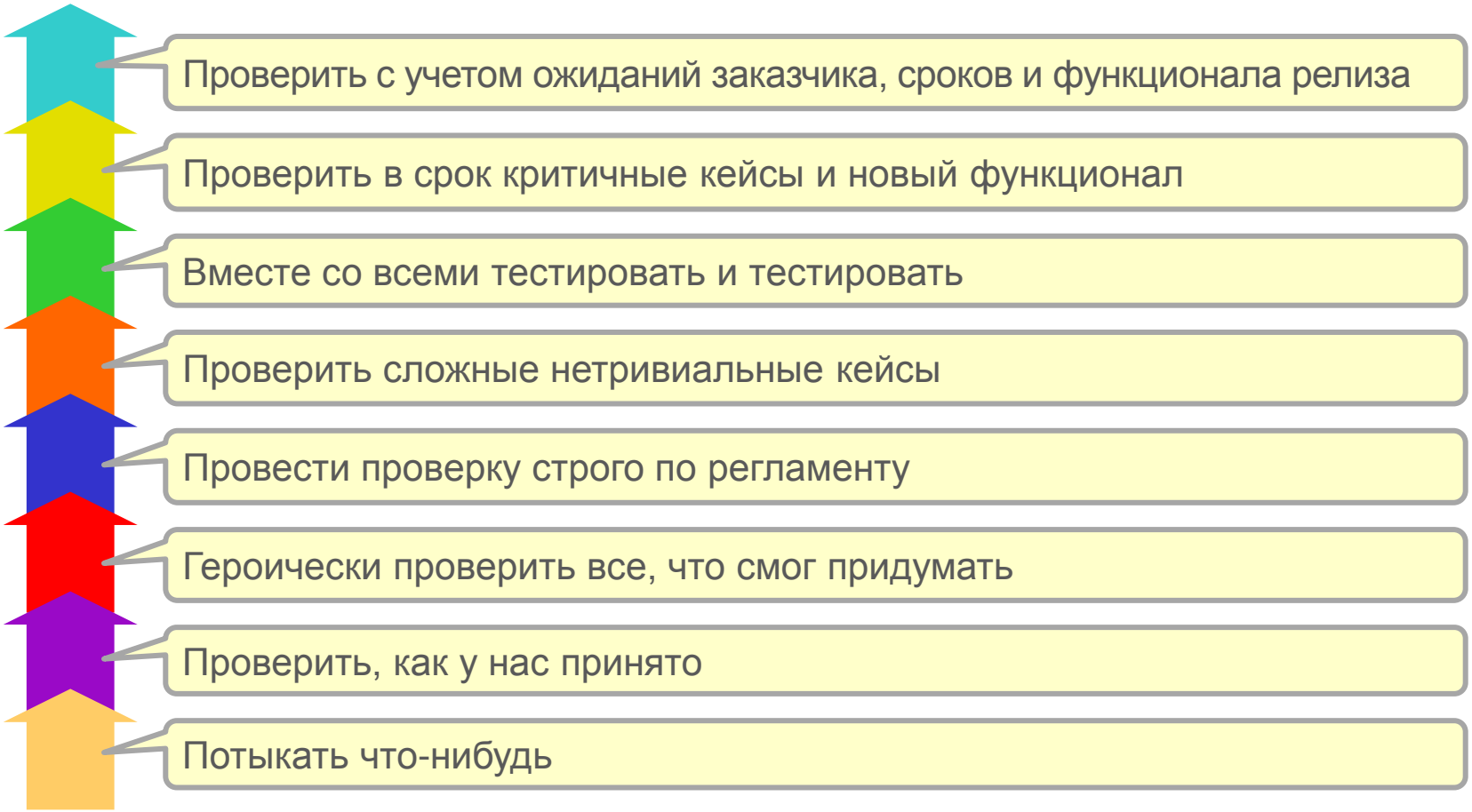
Team

- Каковы ценности и нормы?
- Как организована команда?
- Как организовано взаимодействие?
- Как решаются конфликты?
- Как идет передача ответственности?

Представления о ценностях и нормах образуют устойчивые фреймы.
Их модель дает **Спиральная динамика**,
где выделено **8 уровней**

CUSTIS®

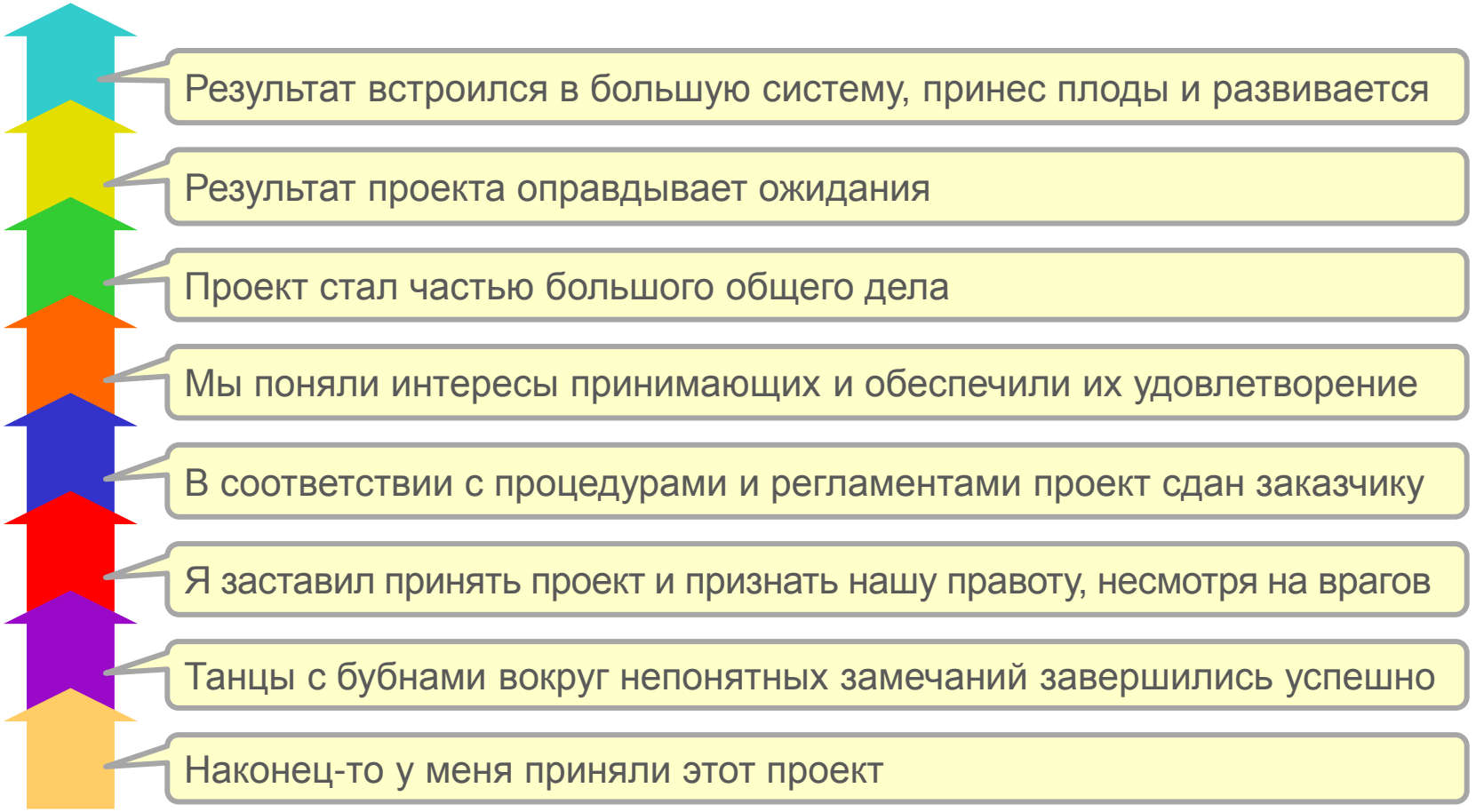
Что значит «протестировать релиз»?



Из доклада [«Действуй, опираясь на ценности»] на AgileDays – 2016

CUSTIS®

Что значит «успешно завершить проект»?



Из доклада «Действуй, опираясь на ценности» на AgileDays – 2016

CUSTIS®

Подводя итоги

- Каждой ИТ-разработке нужно **свое** качество
- Ответственность тоже делится **по-своему**
- Надо договариваться об идеальной картине, учитывая:
  - представления стейкхолдеров и команды
  - объективные особенности проекта
- А затем – работать над воплощением идеала

Спасибо! Вопросы?

Максим Цепков mtsepkov.org